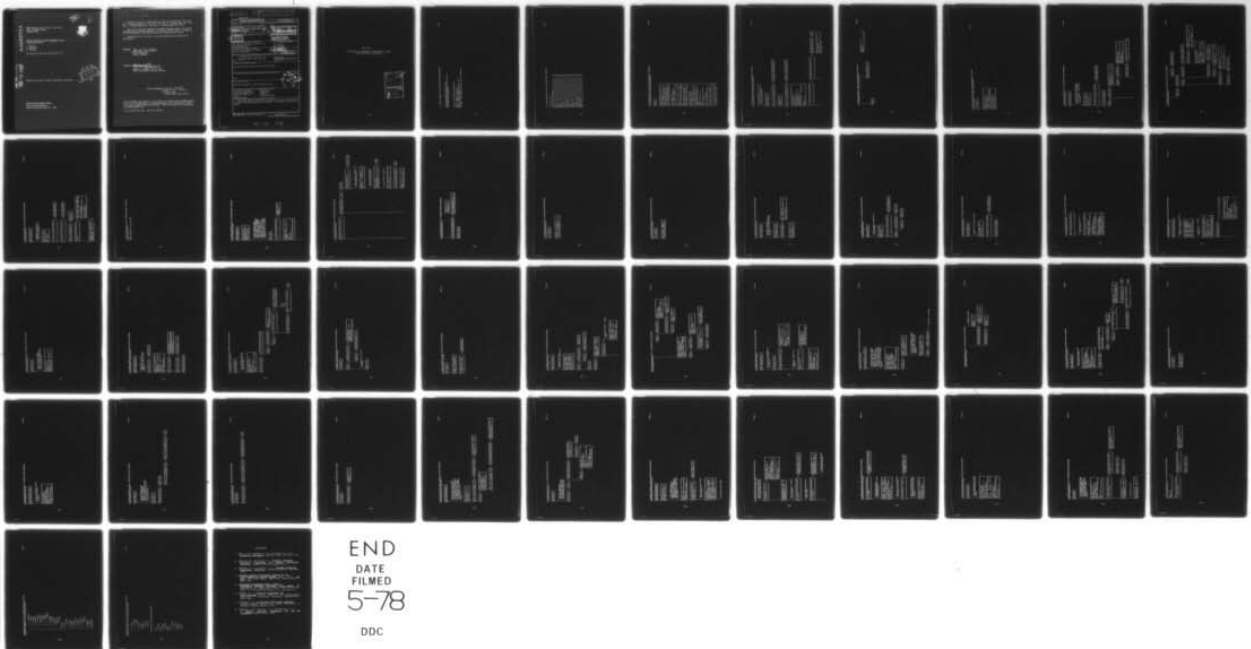


AD-A052 734

CHARLES STARK DRAPER LAB INC CAMBRIDGE MA
JOVIAL STRUCTURED DESIGN DIAGRAMMER (JSDD). VOLUME III. PROGRAM--ETC(U)
FEB 78 G GODDARD, M WHITWORTH, E STROVINK F30602-76-C-0408
R-1120-VOL-3-PT-4 RADC-TR-78-9-VOL-3-PT-4 NL

UNCLASSIFIED

| OF |
AD
A062 734



END
DATE
FILMED
5-78
DDC

AD A 052734

RADC-TR-78-9, Vol III, Part 4 (of four)
Final Technical Report
February 1978



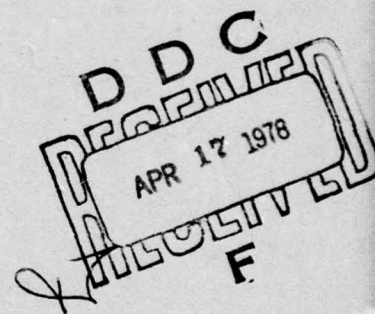
JOVIAL STRUCTURED DESIGN DIAGRAMMER (JSDD), *Volume 4*
Program Description, *Part 4*

G. Goddard
M. Whitworth
E. Strovink

The Charles Stark Draper Laboratory, Inc.

DDC FILE COPY

Approved for public release; distribution unlimited.



ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

Because of the size of this volume, it has been divided into four parts. Part 1 contains pages 1/2 - 123, 649 - 657, Part 2 contains pages 124 - 344. Part 3 contains pages 345 - 592, Part 4 contains pages 593 - 648.

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-78-9, Vol III, Part 4 has been reviewed and is approved for publication.

APPROVED:

Donald Van Alstine
DONALD VANALSTINE
Project Engineer

APPROVED:

Wendall C. Bauman
WENDALL C. BAUMAN, Colonel, USAF
Chief, Information Sciences Division

FOR THE COMMANDER:

John P. Huss
JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (ISIM) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

(18) RADC (19) TR-78-9-VOL-3-PT-4

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RADC-TR-78-9, Vol III, Part 4 (of four)	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) JOVIAL STRUCTURED DESIGN DIAGRAMMER (JSDD) Program Description Volume III. Program Description Part 4.	5. TYPE OF REPORT & PERIOD COVERED Final Technical Report. Sep 76 - Oct 77,	6. PERFORMING ORGANIZATION NUMBER R-1129-VOL-3-PT-4
7. AUTHOR(s) G. Goddard, M. Whitworth E. Strovink	8. PERFORMING ORGANIZATION NAME AND ADDRESS The Charles Stark Draper Laboratory, Inc. 555 Technology Square Cambridge MA 02139	9. SECURITY CLASS. (of this report) UNCLASSIFIED
10. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (ISIM) Griffiss AFB NY 13441	11. REPORT DATE Feb 1978	12. NUMBER OF PAGES 56
13. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Same	14. SECURITY CLASS. (of this report) UNCLASSIFIED	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) Same		
18. SUPPLEMENTARY NOTES RADC Project Engineer: Donald VanAlstine (ISIM)		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Structured programming Preprocessor Structured design diagram Flowcharter Structured extension JOVIAL J3 Parse Invocation diagram Parser generator		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The report presents a detailed description of the JOVIAL Structured Design Diagrammer program implementation for purposes of maintaining or modifying the system.		

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

APR 17 1978
F

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

408 386 JOB

Section 9

Invocation Diagrammer Structured Design
and Invocation Diagrams

ACCESSION for	
NTIS	Write Section <input checked="" type="checkbox"/>
DDC	Buff Section <input type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
JUSTIFICATION	
BY	
DISTRIBUTION/AVAILABILITY CODES	
SPECIAL	
A	23 84-

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER

THIS LISTING CONSISTS OF OUTPUT FROM
THE CHARLES STARK DRAPER LABORATORY'S JOVIAL J3
STRUCTURED DESIGN DIAGRAMMER.

PRINCIPAL DESIGNERS AND IMPLEMENTORS

GARY W. GORDARD, CSOL STAFF
MARK H. WHITWORTH, CSOL STAFF
ERIC F. STROVINK, GRADUATE STUDENT, M.I.T.
COMPUTER SCIENCE DIVISION
THE CHARLES STARK DRAPER LABORATORY, INC., CAMBRIDGE, MA.

C S ORAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER

TABLE OF CONTENTS

THE INVOCATION DIAGRAMMER	3
OUT	7
SUBSTR	9
CAT	11
CHVERT	13
SPACES	18
NULL	19
LENGTH	20
OUT*LINE	21
PUT*OUT	22
STUMP*NO	24
NUMJ	25
INITIALIZATION	26
FIRST*PASS	27
FIND	30
INSERT	32
EQUALS	33
SECOND*PASS	35
SET*BIT	38
GET*BIT	39
WARSHALL	40
CHECK*RECURSION	43
PRINT	44
WRITE*PROC*NAME	45
WRITE*HORIZ	48
PUSH	49

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
DESIGN DIAGRAM OF THE INVOCATION DIAGRAMMER

```

*****
*START $
*****
.
.
*****
*ITEM GTEMP70 H 70 $
*OFFLINE INTEGER --I 36 S-- $
*DEFINE CHARACTER --M 150-- $
*ARRAY BIT*ARRAY 200H 70 $
*ITEM CONT*FLAG INTEGER P 0 $
***TO DETERMINE WHETHER TO
*PRINT CONTINUATION MESSAGE**
*ITEM LINE*NUMBER INTEGER P 0 $
.
*ITEM TEMPH1 H 1 $
*ITEM TEMPH2 CHARACTER $
*ITEM TEMPM3 CHARACTER $
*FILE DIAG*OUT H 10000 V 150 13*
. $
*ITEM PAGE CHARACTER P 5H(PAGE
. $
*ARRAY STOP*FLAG 200 INTEGER $
*ARRAY NEXT*CALLED 200 INTEGER
. $
*ARRAY HOST*PROC 200 INTEGER $
*ARRAY COLUMN 200 INTEGER $
*ITEM HORIZ H 120 P 1H( ) $
*ARRAY FLAG*ARRAY 200 INTEGER $
.
*ITEM INDEX INTEGER $
*ITEM TEMP11 INTEGER $
*ITEM TEMPH30 H 30 $
*FILE FILED B 10000 R 301 V(A)
* V(D) V(C) V(D) V(EOF) 11 $
*ITEM REC*SIZE INTEGER $
*ITEM MAX*ENTRY INTEGER P 0 $
*TABLE FILE08 R 60 S N $
*BEGIN
*ITEM FG*ENTRY H 30 $
*END
*ARRAY PROC*ARRAY 200H 30 $
* BEGIN 3CH(==
) END
*ITEM PTR INTEGER P 0 $
*ITEM STUMP*NUMBER INTEGER P 0
. $
***TO KEEP TRACK OF STUMPS**
*ITEM LINE*LENGTH INTEGER P 118*
. $
***SHOULD EVENTUALLY BE EQUATED*
* TO THE VARIABLE MARK USES TO
* DETERMINE MAY PAGE WIDTH**
*****

```

597

```

PTR = 0 $
PRINT $

```

```

*TEMPI = TEMPI + 1 $

```

*TERM \$ *

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
DESIGN DIAGRAM OF THE INVOCATION DIAGRAMMER

```

*****
* 6 FROM 4 *
*****
.
.
*****
*CONT*FLAG = 1 $
*TEMPH2 = SPACES(1) $
*PUT*OUT(TEMPH2) $
*PUT*OUT(TEMPH2) $
*TEMPH2 = 3*(CONTINUATIONS
* AND INDEPENDENT ROUTINES) $
*PUT*OUT(TEMPH2) $
*TEMPH2 = SPACES(1) $
*PUT*OUT(TEMPH2) $
*PUT*OUT(TEMPH2) $
*****

```

600

601

602

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
DESIGN DIAGRAM OF SUBSTR

```

*ENCODE(CH(I IG), NUM = YC6) *
* *
*TC1 = TC6 $
*OUTCAT(TC, TC1), (PTERR) $
*LN1 = CONST $
*SUBSTR = SF1 $
*RETURN $
*.....

```

```

*.....
*BYTE(86, NUMS)(SF2) = BYTE(85,
* + FIRST, NUMS)(SF1) $
*LN2 = NUM + CONST $
*SUBSTR = SF2 $
*.....

```

C S ORAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
DESIGN DIAGRAM OF CAT

```

.....
*PROC CAT(AA, BB) $ *
.....
*
*
* **RETURNS THE RESULT OF
*   CONCATENATING STRING BB TO
*   THE END OF STRING AA**
*
.....
*ITEM AA M 150 $ *
*ITEM BB M 150 $ *
*ITEM CAT M 150 $ *
*SF1 = AA $ *
*SF2 = BB $ *
.....
*
*
*IF BYTE(10, 18)(SF1) NQ 1H(( ) *
* $ *
*-----SF1 = CNVERT(AA) $ *
*
.....
*
*
*IF BYTE(10, 18)(SF2) NQ 1H(( ) *
* $ *
*-----SF2 = CNVERT(BB) $ *
*
.....
*
*
*IF (LN2 - CONST) EQ 0 $ *-----CAT = SF1 $ *
*-----RETURN $ *
*
.....
*
*
*IF (LN1 + LN2 - CONST2) GR *
* MAXCOL $ *
*-----LN2 = MAXCOL - LN1 + CONST2 $ *
*TC = 20H(*** CONCAT ERROR ***)*
* $ *
*OUT(TC, RPTERR) $ *
*TC = 18H(TRUNCATED STRING) $ *
*
*
*
*OUT(TC, RPTERR) $ *
*OUT(SF2, RPTERR) $ *
*
.....
*
*
*BYTE(16 + LN1 - CONST, LN2 -
* CONST)(SF1) = BYTE(16, LN2 -
* CONST)(SF2) $
*LN1 = LN1 + LN2 - CONST $

```


C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
DESIGN DIAGRAM OF CAT

*CAT = SF1 S

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
DESIGN DIAGRAM OF CNVERT

```

.....
*PROC CNVERT(AA) S *
.....
.
.
.....
*ITEM AA H 158 S *
*ITEM CNVERT H 158 S *
*ITEM DONE I 36 S S *
*ITEM II I 36 S S *
*ITEM JJ I 36 S S *
.....
.
.
.....
**CNVERT CONVERTS NORMAL
** JOVIAL CHARACTER STRINGS OF
** LENGTH 150 TP A DESCRIPTOR
** FORM SUITABLE FOR EXTENDED
** STRING OPERATIONS. CNVERT
** WILL SIMPLY RETURN ITS
** ARGUMENT IF THE STRING IS
** ALREADY CONVERTED. STRINGS
** CONTAINING MORE THA MAXCOL
** CHARACTERS WILL BE TRUNCATED
** .. D ..
.....
.....
*SF3 = AA S *
*.....
.....
*IF BYTE(S8, 18)(SF3) EQ 1H(1) *
* S *
.....
*-----*CNVERT = SF3 S *
*RETURN S *
.....
.
.
.....
*SA3(S8) = 6H( ) S *
***PREVENTS OVERFLOW OF BYTE *
* LAYER **
*II = 1 S *
*DONE = 0 S *
*SF6 = 1H( ) S *
***CLEARS OUT CHARACTER STRING*
*..
.....
.
.
.
.
.
.

```

607

G S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
DESIGN DIAGRAM OF CHVERT

```

.
.
.
.
.
.
.....
*II = II + 1 S *
*LOOP ON II **
.....
.....
.....
*IF II EQ 25 AND DONE EQ 0 S *---SF6 = SPACES(MAXCOL) S *
.....
*LINE WAS ENTIRELY BLANK **
.....
.
.....
*CHVERT = SF6 S *
.....

```


C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
DESIGN DIAGRAM OF CNVERT

```

.....
* 16 FROM 14 *
.....
.
.
.
.....
* BYTE($12 - JJ, 144 - 6 *
* IIS(SF6) = BYTE$II * 6 + 6 *
* 144 - 6 * IIS(SF3) $
.....

```

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
DESIGN DIAGRAM OF CNVERT

```

.....
* 17 FROM 14 *
*.....
*
*
*.....
* LWS = MAXCOL + CONST 8 *
*.. TRUNCATE THE STRING ..*
*.....

```

611

612

613

614

615

616

617

618

619

620

G S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
DESIGN DIAGRAM OF FIRST-PASS

```

*****
* 28 FROM 27 *
*****
.
.
*****
*IFETH *--*FIND(TEMP30) NQ 1 $ *--*INSERT(TEMP30) $
*****
* *--*INSERT NEW NAME IF NOT
* *ALREADY THERE*
* *FLAG*ARRAY(SINDEX + 1$) = 1 $
*****
.
.
.
*****
* *ORIF 1 $ *--*FLAG*ARRAY(SINDEX) = 1 $ *
*****
.
.
* *TO INDICATE THAT PROC IS
* *INTERNAL**
.
*****
*TEMP11 = 2 $ *
*****

```

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
DESIGN DIAGRAM OF FIRST-PASS

```

.....
* 29 FROM 27 *
.....
.
.
.....
*TEMP30 = F0-ENTRY(SI) $ *
**PICKUP SYNTAB ENTRY**
.....
.
.....
*IF FIND(TEMP30) NO 1 $ *
**INSERT IF NOT THERE**
*-----*INSERT(TEMP30) $ *
.....

```

623

624

625


```
*****
**PROC EQUALS(SEARCH, TEMP30) &
*****
```

```

* *COMPARES CHARACTER STRINGS
* *ON A BIT LEVEL FOR SORTING
* *PURPOSES. RETURNS 0 IF
* *ARGUMENTS ARE EQUAL (IF
* *INDEXED NAME IN PROC=ARRAY
* *(INDEXED BY SEARCH) IS EQUAL
* *TO ARGUMENT STRING TEMP30). 1
* *IF TEMP30 BELONGS AFTER THE
* *INDEX, AND -1 IF IT BELONGS
* *BEFORE*.

```

[illegible]

```

*****
***** DO WHILE {1} *****
***** *-----*COUNT = COUNT + 1 *
***** *TEMP1 = BIT(COUNT * 30, *
***** * 30)(TEMP3) *
*****
*****

```

- ••CLEARLY, COMPARISON IS GOING
- TO TAKE PLACE ON BIT LEVEL,
- SINCE WE HAVE TO TREAT THE
- CHARACTERS IN THE NAMES AS
- NUMBERS..

```
TEMP12 = BIT(COUNTER * 30,
              30)(TEMP30) $
```

```

*****      *****
**IFEITH **--+TEMPI LS TEMPI2 $ +---+
*****      *****
**WE'RE LUCKY - THE FIRST

```

```

*****
* COMPARISON WORKED **
*
*****
*-----*
*EQUALS = - 1 $ *
*RETURN $      *
*****
*****
*****ORIF TEMPI1 GR TEMPI2 $ -----*
*****
*****SAKE LUCK AS ABOVE**
*****
*****RETURN $ *****
*****
*****
*****ORIF COUNTER EQ 5 $ -----*
*****
*****EQUALS = 0 $ *
*****RETURN $ *****
*****

```

628

G S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
DESIGN DIAGRAM OF SECOND-PASS

```

.....
* 36 FROM 35 *
.....
.
.
.....
*CUR*PROC = INDEX $ *
**LOOKUP NAME**
.....

```

[illegible]

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
DESIGN DIAGRAM OF SET-BIT

```

*****
*PROC SET-BIT(AA, BB, CC) $ *
*****
.
.
.
*..UTILITY PROCEDURE TO SET A
* BIT IN THE BIT ARRAY
* DATABASE..
.
.
*****
*ITEM AA INTEGER $
*ITEM BB INTEGER $
*ITEM CC INTEGER $
*ITEM TEMP70 W 70 $
*TEMP70 = BIT-ARRAY(AAAS) $
*****
.
.
.
*..INTRODUCE TEMPORARY TO AVOID
* COMPILER BOMB..
.
*****
*BIT(88 * 2, 28)(TEMP70) =
* BIT(84, 28)(CC) $
*****
.
.
.
*..BIT-ARRAY IS IMPLEMENTED AS
* CHARACTER STRINGS, SO THAT $
* WHY THE BIT FUNCTION MUST BE
* USED..
.
.
*****
*BIT-ARRAY(AAAS) = TEMP70 $
*..RE-ASSIGNMENT OF TEMPORARY..
*
*****

```

632

```
*****  
**PROC MARSHALL $ **  
*****  
. . .  
. . .  
. . .  
..THIS PROCEDURE IS NOT  
..MARSHALL'S ALGORITHM.  
..HOWEVER, IT DOES DO A  
..TRANSITIVE CLOSURE ON THE BIT  
..MATRIX, ASSIGNING NEW VALUE  
..OF "2" TO N-LEVEL  
..TRANSITIVITY, N GR 1..  
..  
.....  
*ITEM FLAG B $ *  
*FLAG = 0 $ *  
.....  
..  
.....  
*DO WHILE [NOT FLAG] .....  
.....  
.....
```

```
*****  
      41 FROM 68  
*****  
. .  
*****  
      IF GET'BIT(J, I) NQ 0 $ *****  
      FOR K = 0, MAX-ENTRY - 1 $ *****  
      IF GET'BIT(I, K) NQ 0 $ ***** 42
```

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
DESIGN DIAGRAM OF MARSHALL

```

.....
* 42 FROM 41 *
.....
.
.
.
.....
*IF GET-BIT(J, K) EQ 8 $ *-----SET-BIT(J, K, 2) $ *
.....
*FLAG = 0 $
.....

```



```
*****PROC CHECK* RECURSION $*****
```

- "ALL WE HAVE TO DO TO CHECK
- FOR RECURSIVE LOOPS IS TO
- EXAMINE THE MAIN DIAGONAL OF
- THE BIT MATRIX. IF THERE ARE
- ANY NONZERO ELEMENTS, THEN
- THE PROCEDURE OF THAT INDEX
- IS EITHER PART OF A RECURSIVE
- LOOP OR SIMPLY CALLS
- ITSELF."

```
*****
*ITEM FLAG B $
*ITEM MESSAGE M 150 $
*FLAG = 0 $
*****
```

```
*****
**FOR I = 0, 1, MAXENTRY - 1 $ **---IF GET'DIIL, I) HQ 0 $ *****
*****
***** *FLAG*ARRAY(SIS) = 10000 $ *
***** *FLAG = 1 $ *****
```

```
*****
**IF FLAG $ *****
**MESSAGE = 25H(ULTIMATELY-
**  SELF-RECURSIVE) $
**PUT*OUT(MESSAGE) $
**MESSAGE = SPACES(1) $
**PUT*OUT(MESSAGE) $
*****
```

```
*****  
FOR I = 0, 1, MAXENTRY - 1 $ *****  
      IF FLAGARRAY(SIS) EQ 1000 $  
          MESSAGE = 'PROC' ARRAY(SIS) $  
          PUT 'OUT( MESSAGE ) $  
          FLAG  
*****
```

637

C.3 DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
DESIGN DIAGRAM OF WRITE*PROC*NAME

```

*****
*PROC WRITE*PROC*NAME S *
*****
.
.
*****
*ITEM AA N 158 S *
*ITEM TEMP11 INTEGER S *
*ITEM HOST*P N 38 S *
*****
.
.
*..THIS PROCEDURE DOES
*   FORMATTING OF HORIZ AND
*   WRITES OUT A RIGHTMOST
*   PROCEDURE NAME. UNNECESSARY
*   COMPLICATION DUE TO USE OF
*   BYTE FUNCTION TO FORMAT LINES
*   FOR OUTPUT..
.
.
*****
*TEMP11 = 0 S
*HOST*P = PROC*ARRAY(SHOST*PROC*
*(SPTR81) S
*..PICKUP NAME OF CURRENT..
*****
.
.
*****
*DO WHILE (BYTE(STEMP11,
* 18)(HOST*P) NQ 1M ) AND *---*TEMP11 = TEMP11 + 1 S *
*  TEMP11 LQ 29)
*****
.
.
*..NOW WE HAVE LENGTH OF PROC
*   NAME IN TEMP11..
.
.
*****
*AA = HORIZ S
*..HORIZ CONTAINS THE VERTICAL
*   LINES IN THE INV. DIAGRAM..
*   BYTE(SCOLUMN(SPTR - 18) + 1.
* 28)(AA) = 2M(---) S
*..MAKE A HORIZONTAL BRANCH OFF
*   FROM THE RIGHTMOST EDGE OF
*   THE OLD HORIZ..
*****
.
.
*..FOLLOWING PIECE OF CODE

```

[illegible]

640

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
DESIGN DIAGRAM OF WRITE*HORIZ

```

*****
*PROC WRITE*HORIZ $
*****
.
.
.
*..THIS PROCEDURE DUMPS HORIZ
*  BASED ON THE CURRENT COLUMN
*  VALUE..
.
*****
*ITEM TEMPC1 H 150 $
*ITEM R1 INTEGER $
*R1 = COLUMN(PT8) + 1 $
*TEMPC1 = 1H( ) $
*BYTE(80, R1)(TEMPC1) =
*  BYTE(80, R1)(HORIZ) $
*TEMPH1 = BYTE(80, 15)(TEMPC1)
* $
*BYTE(80, 15)(TEMPC1) = 1H( ) $
* $
*TEMPH2 = BYTE(80,
*  R1)(TEMPC1) $
*TEMPC1 = CONVERT(TEMPH2) $
*BYTE(86, 15)(TEMPC1) = TEMPH1
* $
*PUT*OUT(TEMPC1) $
*****

```

642

C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
DESIGN DIAGRAM OF PUSH

```

*****
* DONE = 0 $
* TEMPI1 = 0 $
*****
.
*****
* DO WHILE (TEMPI1 LQ MAX*ENTRY
* - 1 AND NOT DONE)
*****
*-----IF GET*BIT(HOST*PROC($PTRS)).
* TEMPI1 EQ 1 $
*****
*-----NEXT*CALLED($PTRS) = TEMPI1 $
* STOP*FLAG($PTRS) = 0 $
* DONE = 1 $
*****
.
*****
* TEMPI1 = TEMPI1 + 1 $
*****

```

ULTIMATELY SELF-RECURSIVE

**CONVERT
OUT
SPACES**

```

---MAIN
*---CHECK*RECURSION
*
*---GET*BIT
*
*---PUT*OUT
*
*---CAT
*
*---CNVERT
*
*---SPACES
*
*---OUT
*
*---CNVERT*
*
*---TRMOUT+
*
*
*---OUT*
*
*---ENCODE+
*
*---LENGTH
*
*---CNVERT*
*
*---OUT*LINE
*
*---CNVERT*
*
*---LENGTH
*
*---CNVERT*
*
*---REMQUO+
*
*---SPACES*
*
*---SPACES*
*
*---FIRST*PASS
*
*---FIND
*
*---EQUALS
*
*---REMQUO+
*

```


C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
INVOCATION DIAGRAM OF THE INVOCATION DIAGRAMMER

```

.  *--INSERT
.  *--INITIALIZATION
.  *--CAT
.  *--CNVERT*
.  *--OUT*
.  *--CNVERT*
.  *--PRINT
.  *--PUSH
.  *--GET*BIT
.  *--WRITE*HORIZ
.  *--CNVERT*
.  *--PUT*OUT
.  *--CAT
.  *--CNVERT*
.  *--OUT*
.  *--ENCODE*
.  *--LENGTH
.  *--CNVERT*
.  *--OUT*LINE
.  *--CNVERT*
.  *--LENGTH
.  *--CNVERT*
.  *--REMQUO*
.  *--SPACES*
.  *--WRITE*PROC*NAME
.  *--CNVERT*
.  *--PUT*OUT
.  *--CAT

```


C S DRAPER LABORATORY JOVIAL STRUCTURED DESIGN DIAGRAMMER
INVOCATION DIAGRAM OF THE INVOCATION DIAGRAMMER

```

. *--FIND
. .
. *--EQUALS
. .
. *--REMQUO+
. .
. *--SET*BIT
. .
. *--SPACES+
. .
. *--MARSHALL
. .
. *--GET*BIT
. .
. *--SET*BIT

```

CONTINUATIONS AND INDEPENDENT ROUTINES

```

--NULL
.
. *--CNVERT+
--NUMJ
.
. *--ENCODE+
.
. *--TRMOUT+
--SUBSTR
.
. *--CAT
.
. *--CNVERT+
.
. *--OUT+
.
. *--CNVERT+
.
. *--ENCODE+
.
. *--OUT+

```

REFERENCES

1. Dahl, O.-J., Dijkstra, E. W., and Hoare, C. A. R., Structured Programming, Academic Press, New York, 1972.
2. Hamilton, M., and Zeldin, S., Top-Down, Bottom-Up Structured Programming and Program Structuring, (Revision 1), CSDL - E-2728, December 1972.
3. McGowan, C. L., and Kelly, J. R., Top-Down Structured Programming Techniques, Petrocelli/Charter, New York, 1975.
4. Standard Computer Programming Language for Air Force Command and Control Systems, Short Title: CED 2400, Air Force Manual AFM 100-24, Reprint dated 21 April 1972.
5. Structured Programming Series, Volume I, Programming Languages Standards, Final Report, IBM Corporation for RADC under contract F30602-74-C-0186, March 15, 1975, RADC-TR-74-300, Vol I, AD# A016 771.
6. DeRemer, F. L., Practical Translators for LR(k) Languages, Ph.D. Thesis at the Massachusetts Institute of Technology, Cambridge, Massachusetts, Sept. 1969.
7. Lalonde, W. R., An Efficient LALR Parser Generator, Technical Report CSRG-2, M.Sc. Thesis, University of Toronto, Toronto, Ontario, 1970.
8. McKeeman, W. M., Horning, J. J., and Wortman, D. B., A Compiler Generator Implemented for the IBM System/360, Prentice Hall, 1970.